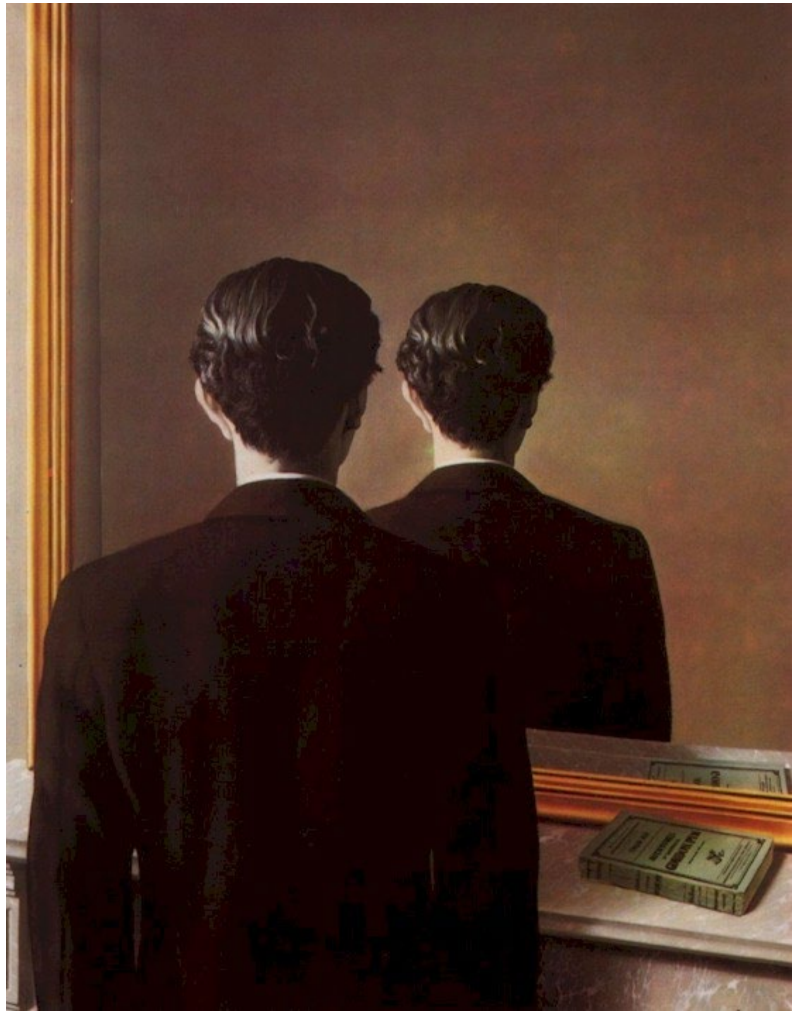
3D Modeling for a Single View



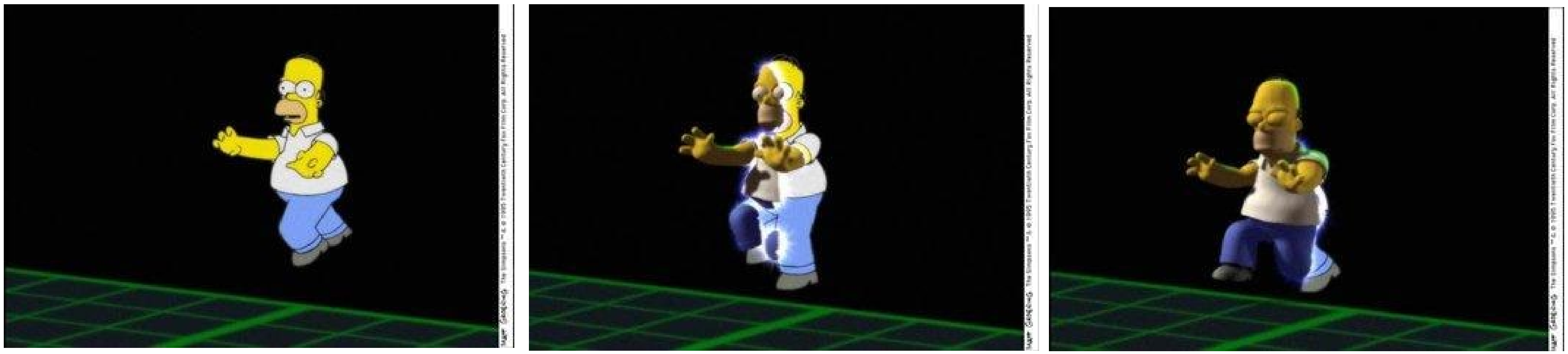
René MAGRITTE
Portrait d'Edward James

CS180: Intro to Comp. Vision and Comp. Photo
Alexei Efros, UC Berkeley, Fall 2024

…with a lot of slides stolen from Steve Seitz and David Brogan,

Breaking out of 2D

…now we are ready to break out of 2D



And enter the real world!

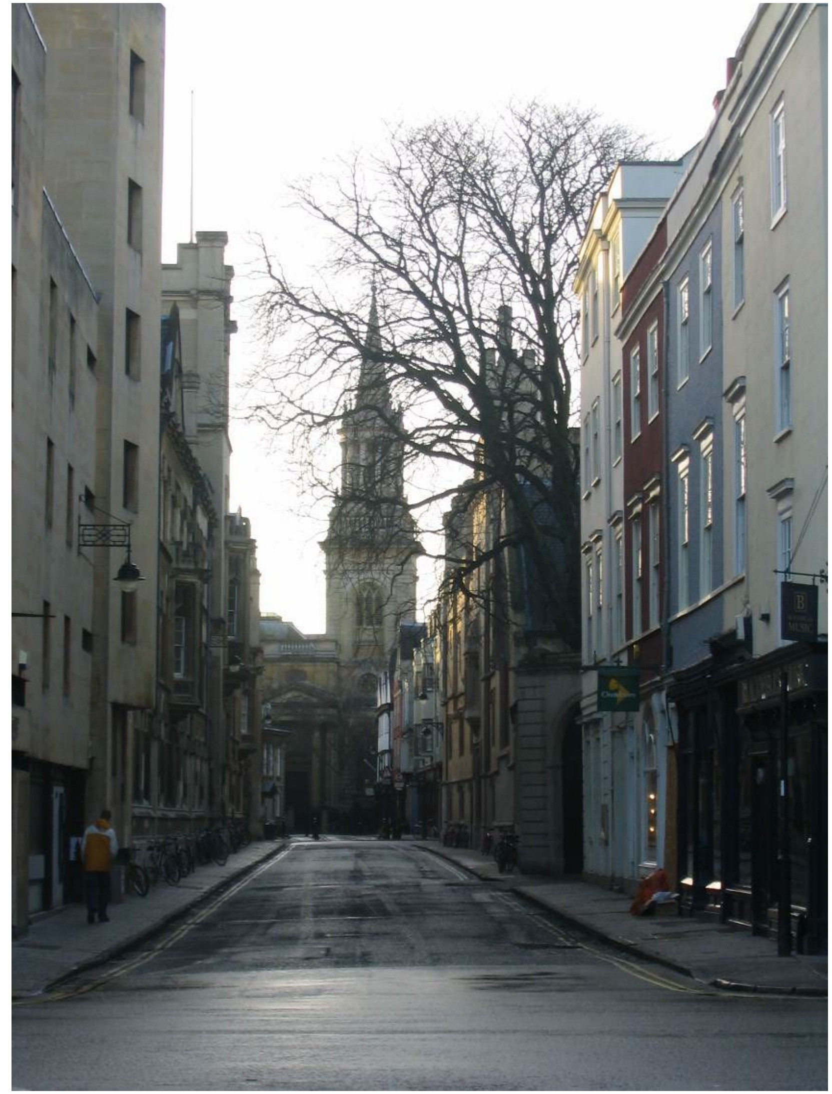
on to 3D…

Enough of images!

We want more of the
plenoptic function

We want real 3D scene
walk-throughs:
- Camera rotation
- Camera translation

Can we do it from a single
photograph?



Camera rotations with homographies

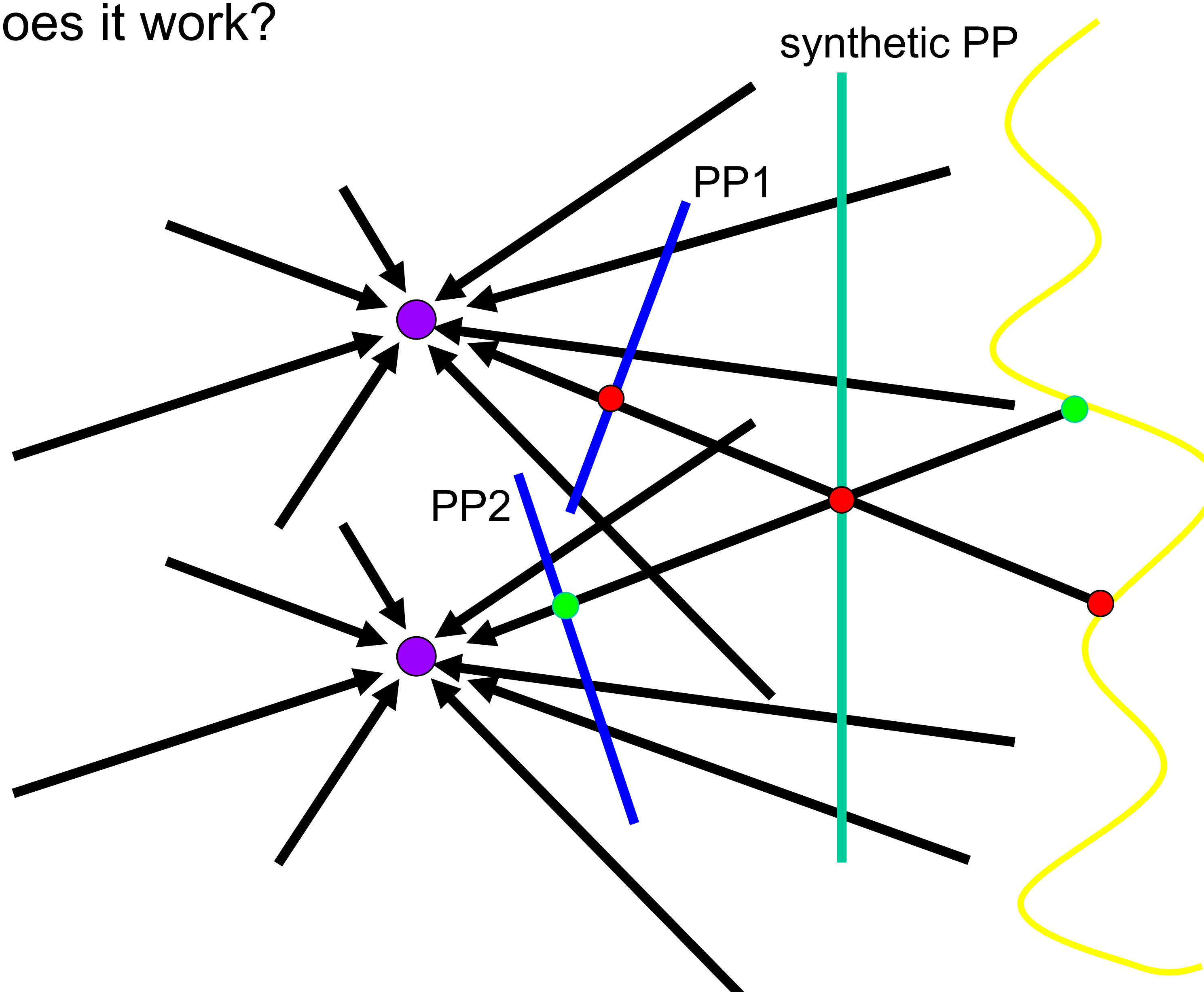
Original image

St.Petersburg
photo by A. Tikhonov

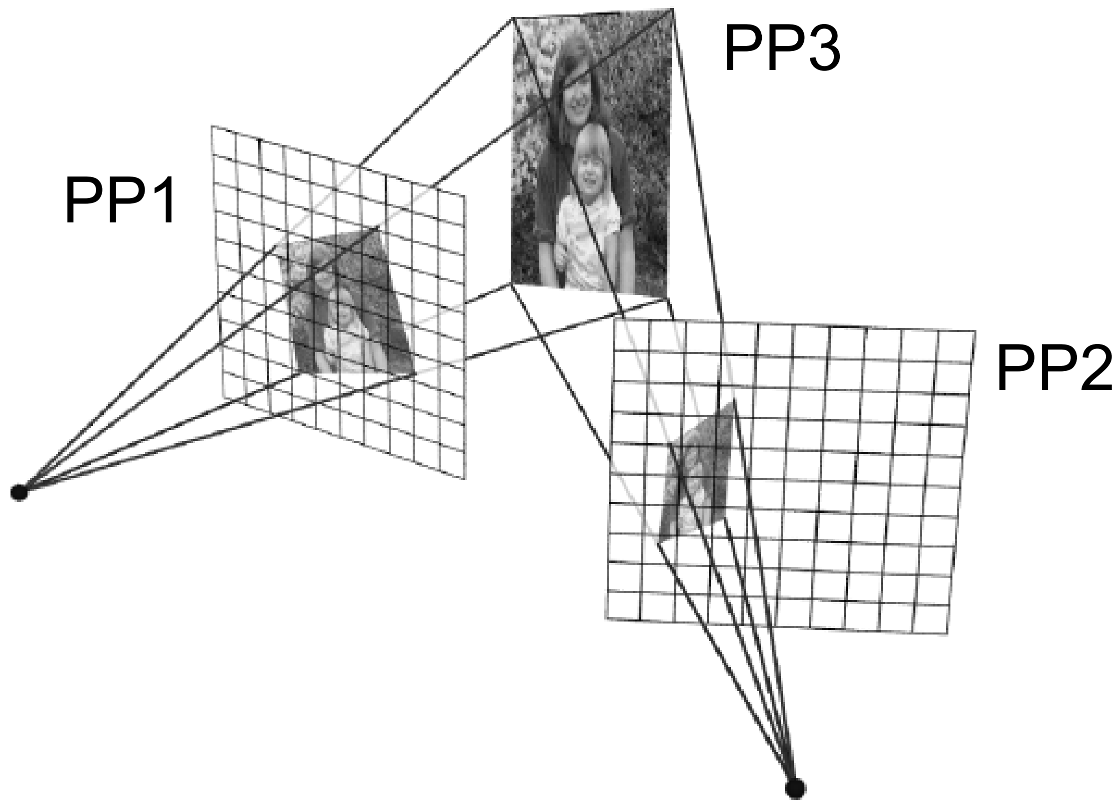
Virtual camera rotations

Camera translation

Does it work?



Yes, with planar scene (or far away)



PP3 is a projection plane of both centers of projection, so we are OK!

So, what can we do here?

Model the scene as a set of planes!

Now, just need to find the orientations of these planes.

Automatic Photo Pop-up

Some preliminaries: projective geometry

Ames Room

Silly Euclid!

Parallel lines???

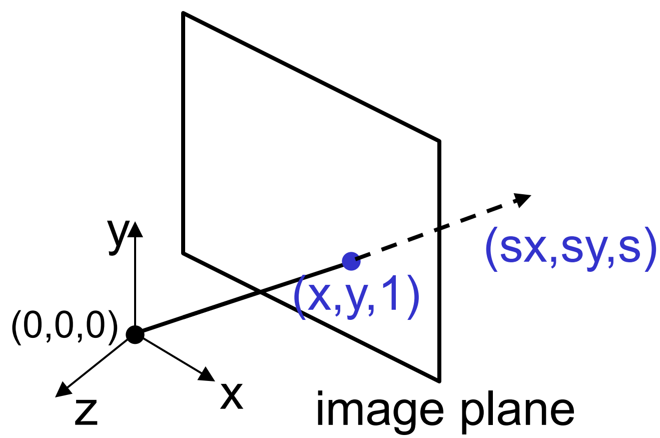
The projective plane

Why do we need homogeneous coordinates?

- represent points at infinity, homographies, perspective projection, multi-view relationships

What is the geometric intuition?

- a point in the image is a *ray* in projective space

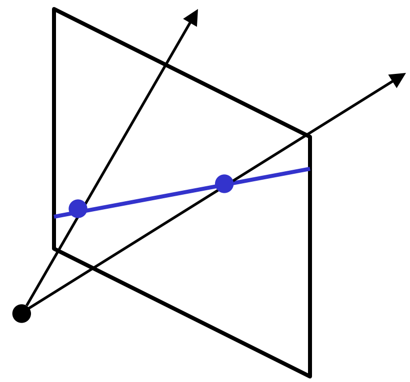


- Each *point* (x,y) on the plane is represented by a *ray* (sx,sy,s)
  - all points on the ray are equivalent: (x, y, 1) ≅ (sx, sy, s)

Remember projection eq: $(x, y, z) \rightarrow (-d\frac{x}{z},\ -d\frac{y}{z})$

Projective lines

What does a line in the image correspond to in projective space?

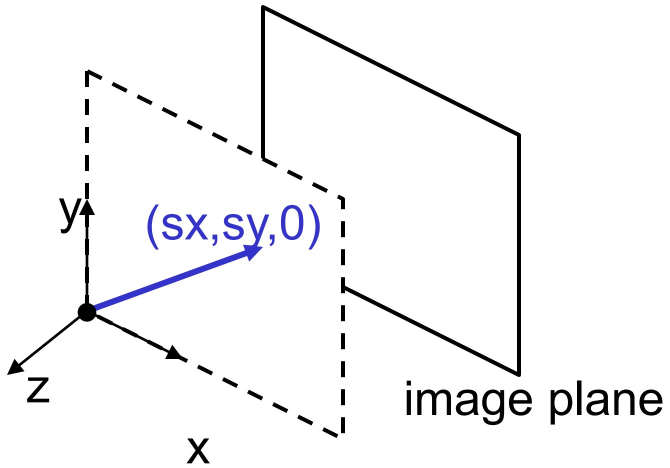


- A line is a *plane* of rays through origin
  - all rays (x,y,z) satisfying: ax + by + cz = 0

$$\text{in vector notation:} \qquad 0 = \underbrace{\begin{bmatrix} a & b & c \end{bmatrix}}_{\mathbf{l}} \underbrace{\begin{bmatrix} x \\ y \\ z \end{bmatrix}}_{\mathbf{p}}$$

- A line is also represented as a homogeneous 3-vector **l**

Ideal points and lines



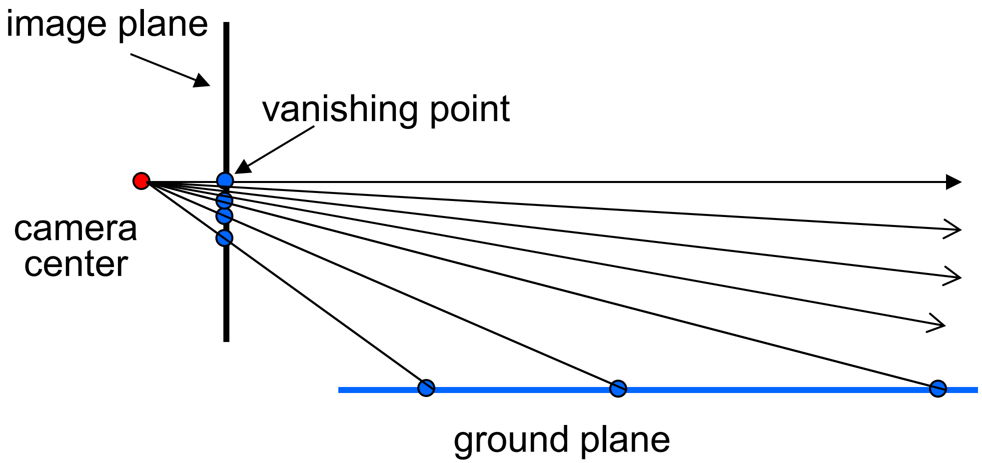
Ideal point (“point at infinity”)

- p ≅ (x, y, 0) – parallel to image plane
- It has infinite image coordinates

Ideal line

- l ≅ (0, 0, 1) – parallel to image plane

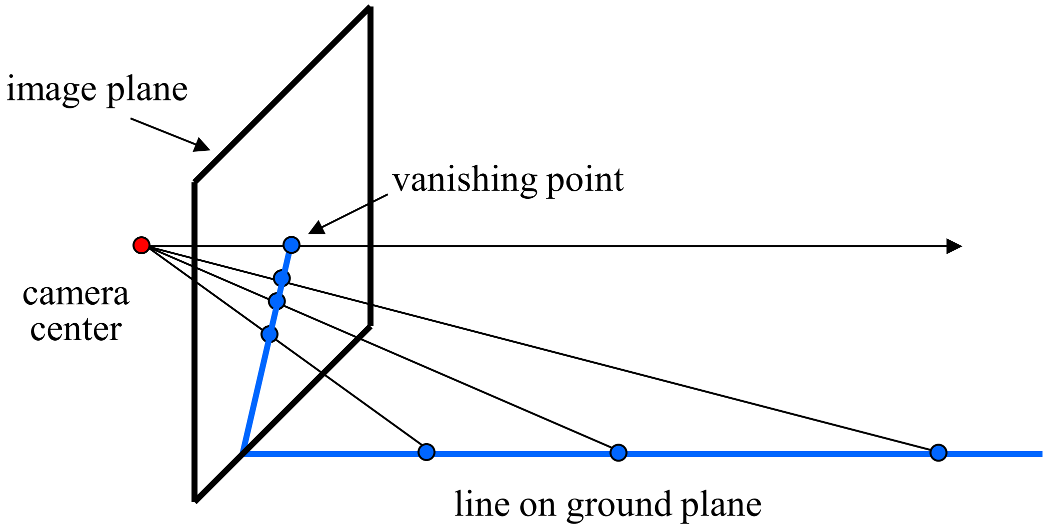
Vanishing points



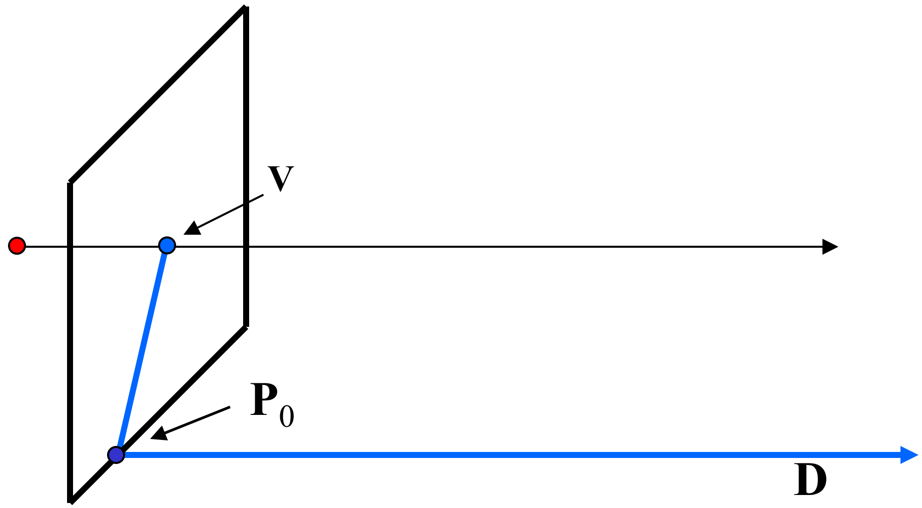
Vanishing point

- projection of a point at infinity

Vanishing points (2D)



Computing vanishing points



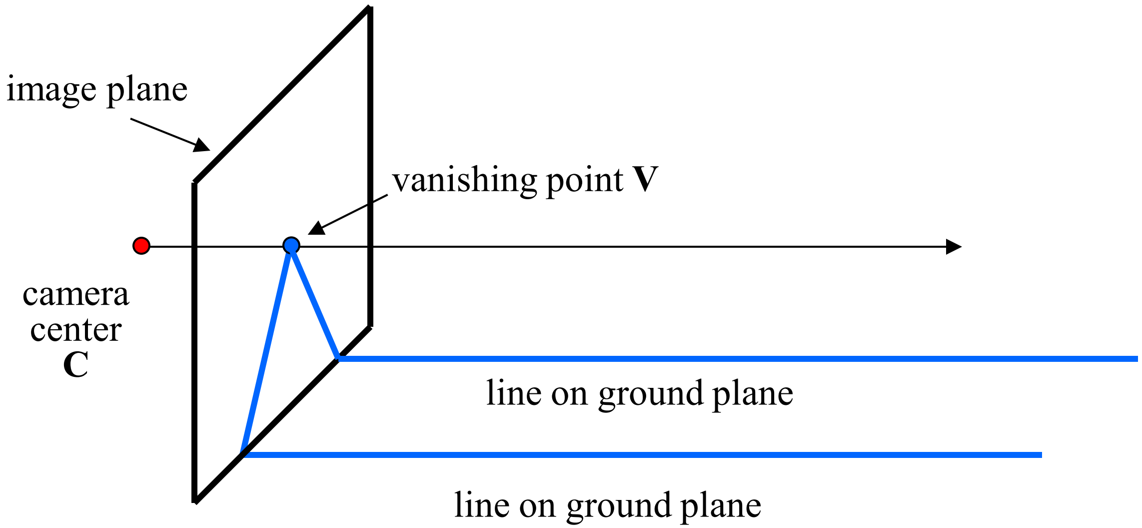
$$\mathbf{P} = \mathbf{P}_0 + t\mathbf{D}$$

$$\mathbf{P}_t = \begin{bmatrix} P_X + tD_X \\ P_Y + tD_Y \\ P_Z + tD_Z \\ 1 \end{bmatrix} \cong \begin{bmatrix} P_X / t + D_X \\ P_Y / t + D_Y \\ P_Z / t + D_Z \\ 1/t \end{bmatrix} \qquad t \to \infty \qquad \mathbf{P}_\infty \cong \begin{bmatrix} D_X \\ D_Y \\ D_Z \\ 0 \end{bmatrix}$$

Properties $\quad \mathbf{v} = \mathbf{\Pi}\mathbf{P}_\infty$

- $\mathbf{P}_\infty$ is a point at *infinity*, **v** is its projection
- They depend only on line *direction*
- Parallel lines $\mathbf{P}_0$ + t**D**, $\mathbf{P}_1$ + t**D** intersect at $\mathbf{P}_\infty$

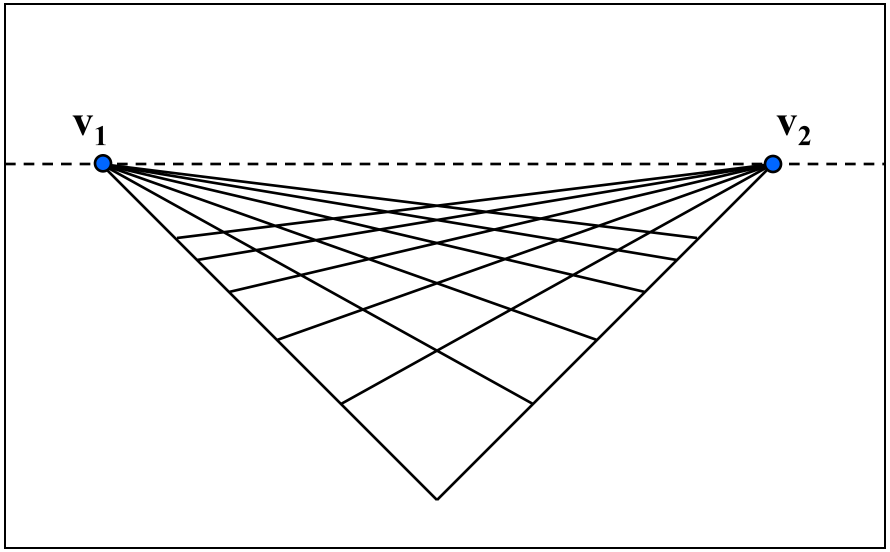
Vanishing points



Properties

- Any two parallel lines have the same vanishing point **v**
- The ray from **C** through **v** is parallel to the lines
- An image may have more than one vanishing point

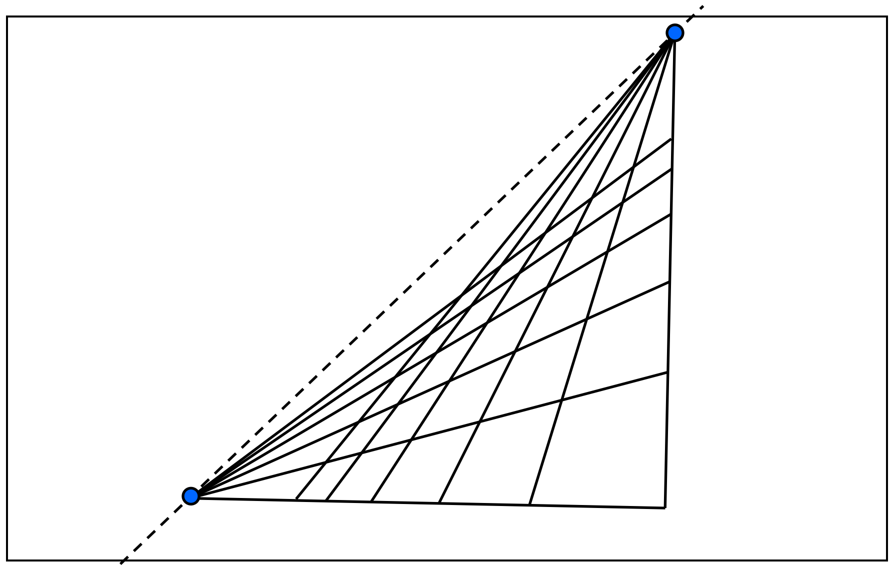
Vanishing lines



Multiple Vanishing Points

- Any set of parallel lines on the plane define a vanishing point
- The union of all of these vanishing points is the *horizon line*
  – also called *vanishing line*
- Note that different planes define different vanishing lines

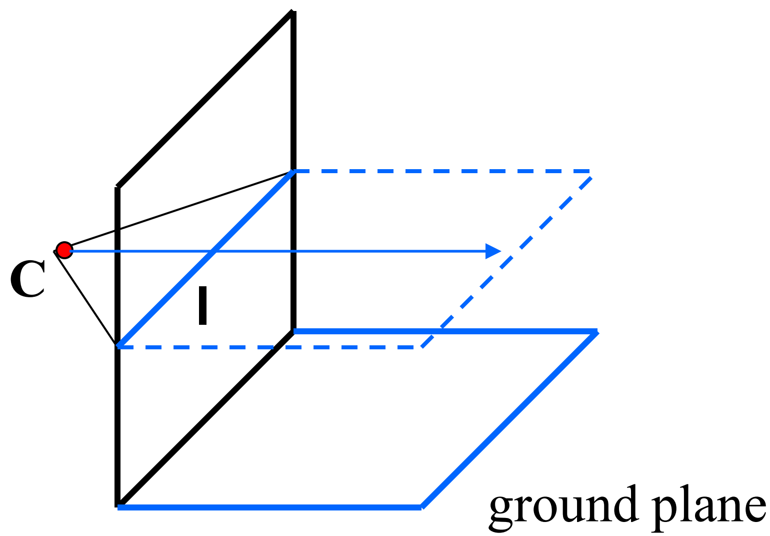
Vanishing lines



Multiple Vanishing Points

- Any set of parallel lines on the plane define a vanishing point
- The union of all of these vanishing points is the *horizon line*
  - also called *vanishing line*
- Note that different planes define different vanishing lines

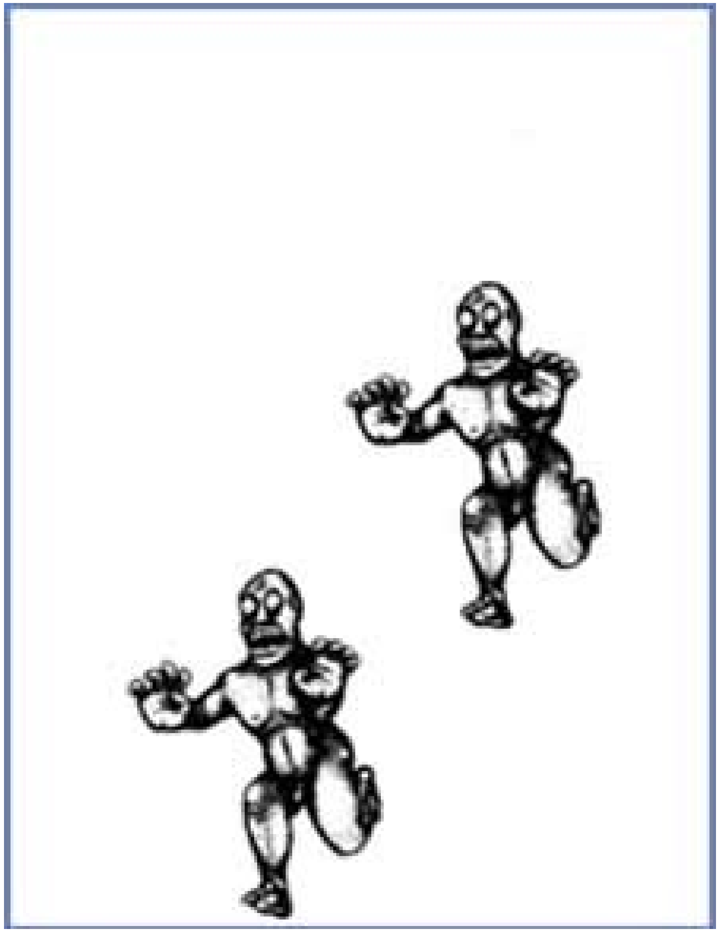
Computing vanishing lines

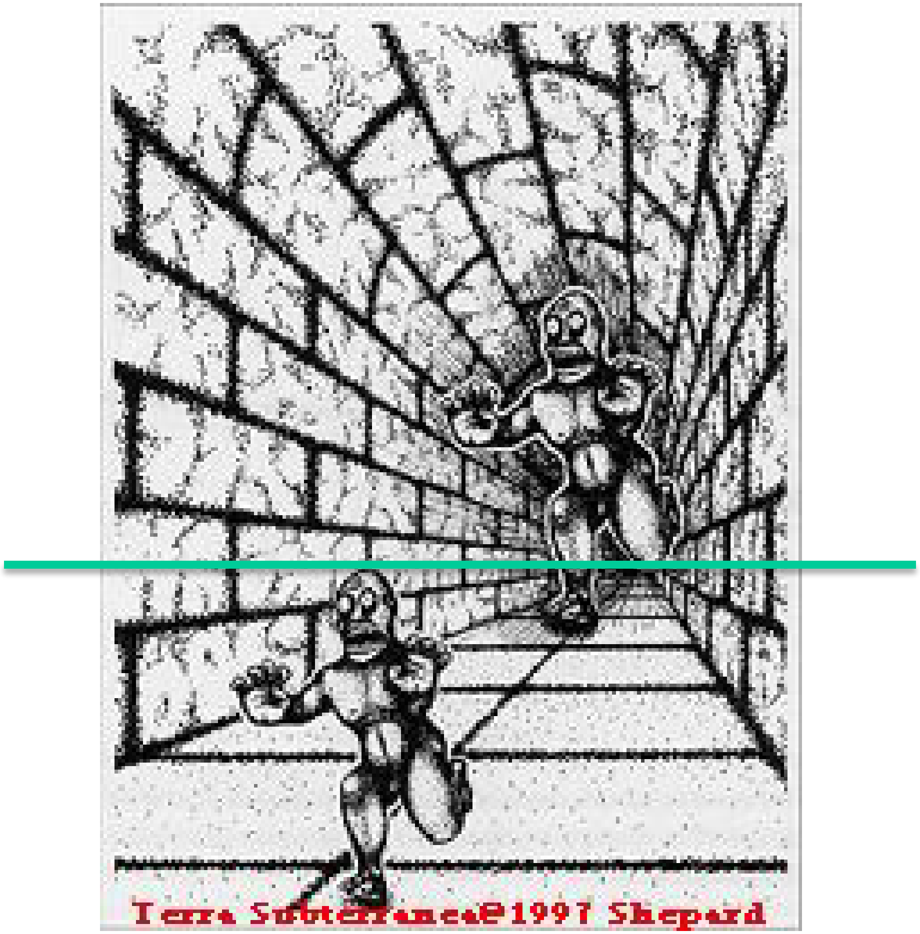


Properties

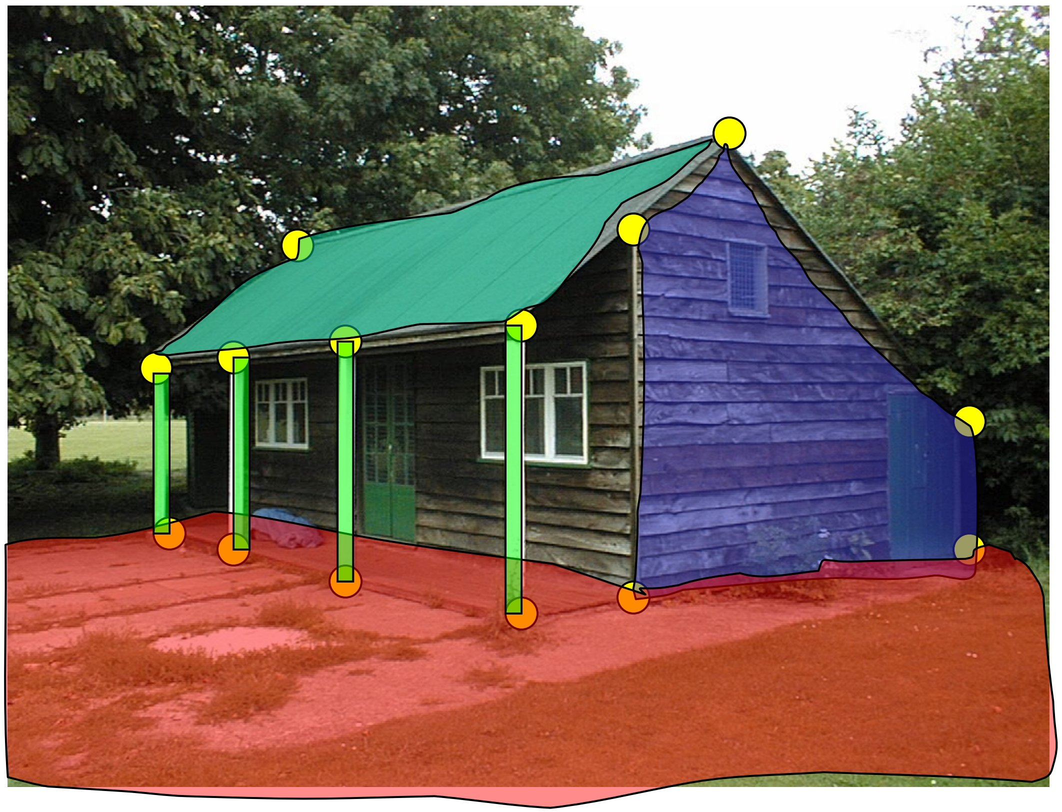
- **l** is intersection of horizontal plane through **C** with image plane
- Compute **l** from two sets of parallel lines on ground plane
- All points at same height as **C** project to **l**
  - points higher than C project above l
- Provides way of comparing height of objects in the scene

Fun with vanishing points



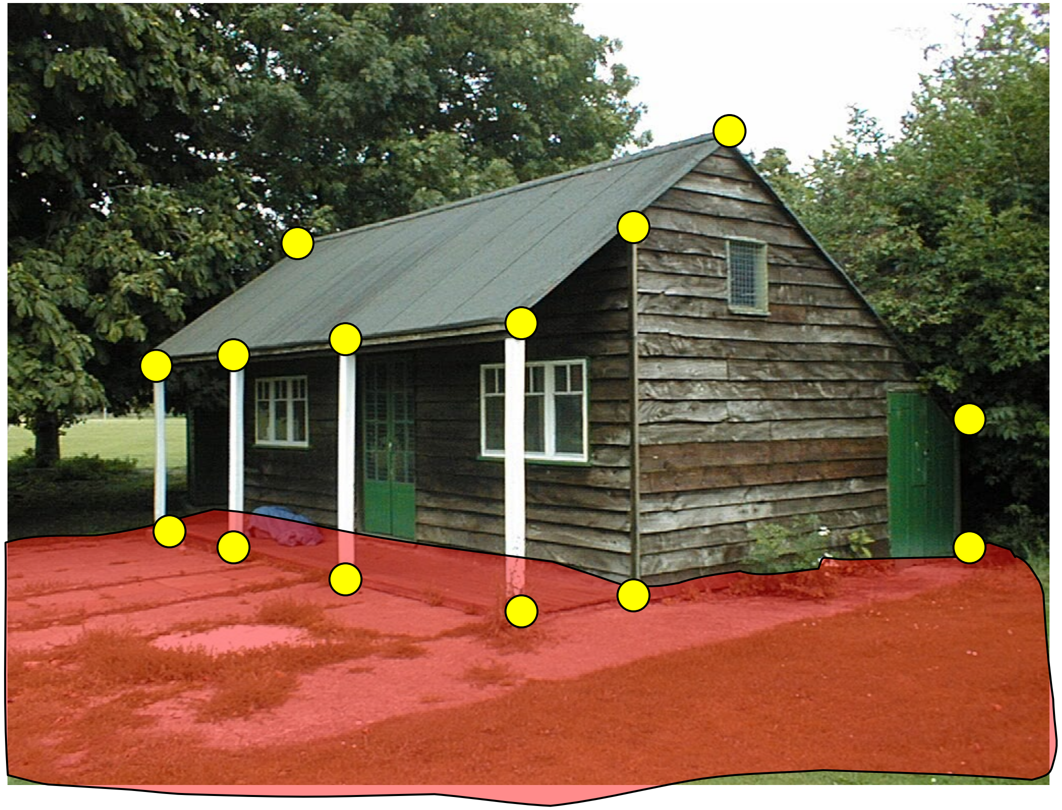


3D from single image



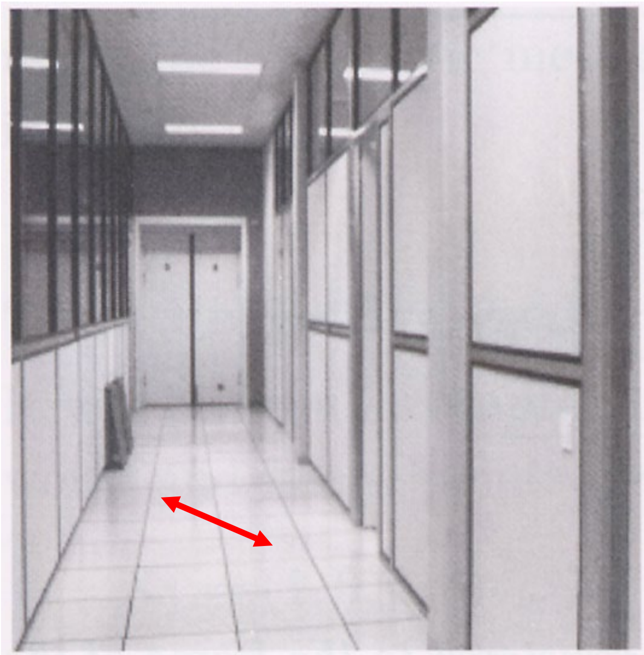
1. Find world coordinates (X,Y,Z) for a few points
2. Connect the points with planes to model geometry
   - Texture map the planes

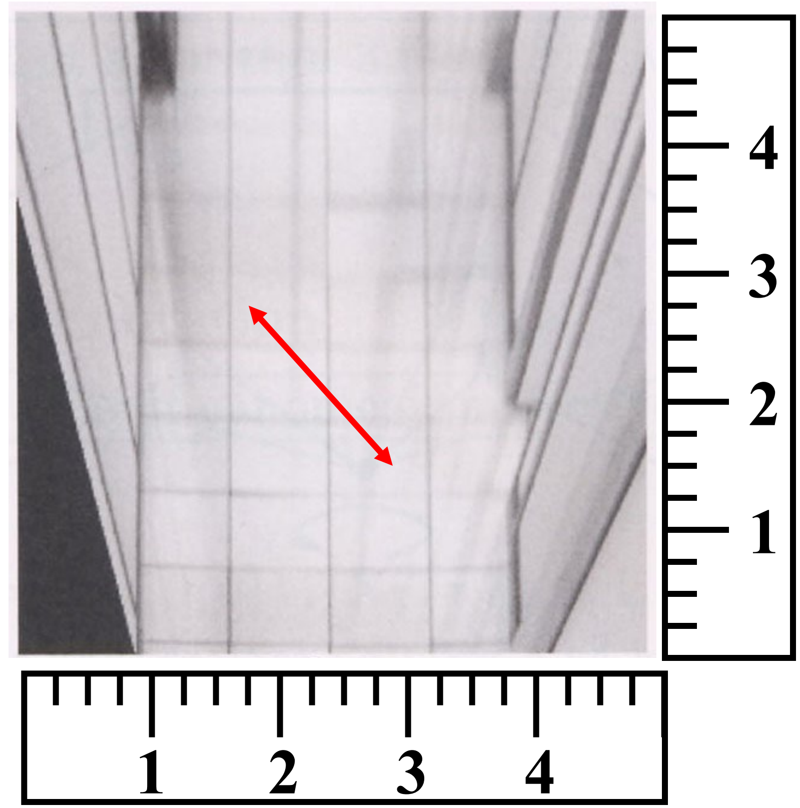
Finding world coordinates (X,Y,Z)



1. Define the ground plane (Z=0)
2. Compute points (X,Y,0) on that plane
3. Compute the *heights* Z of all other points

Measurements on planes

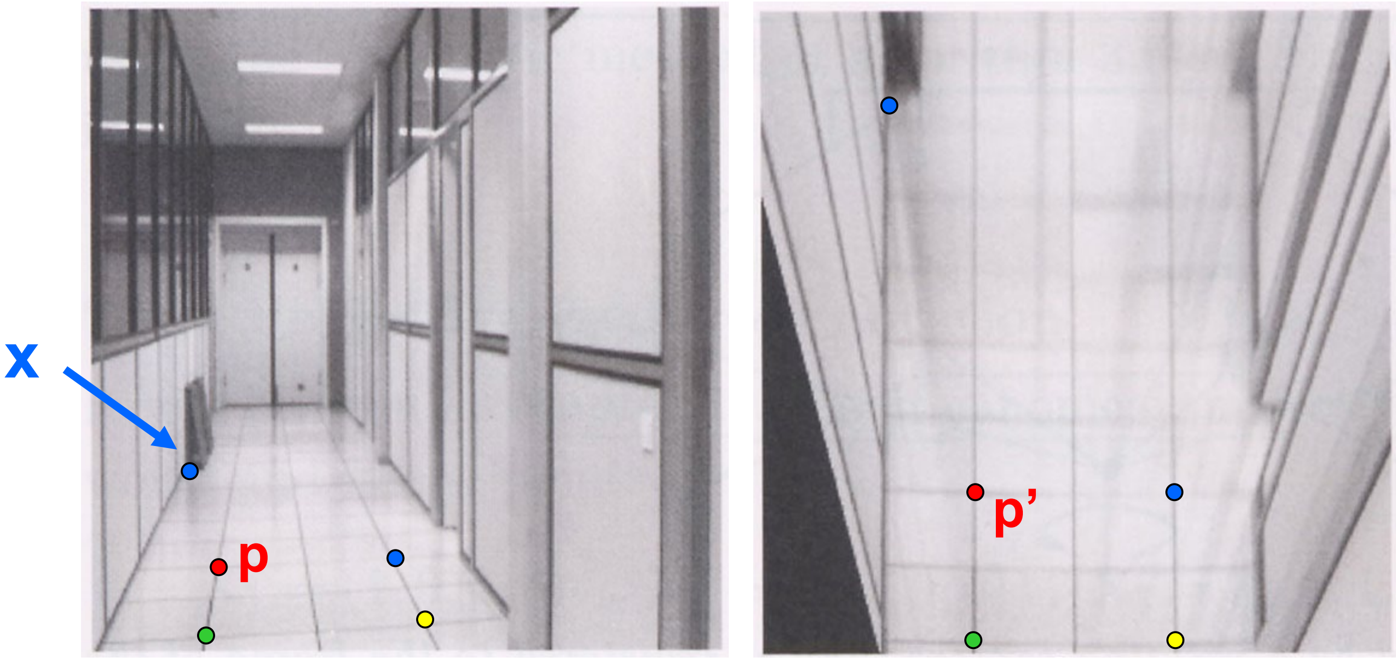




Approach: unwarp, then measure

What kind of warp is this?

Unwarp ground plane



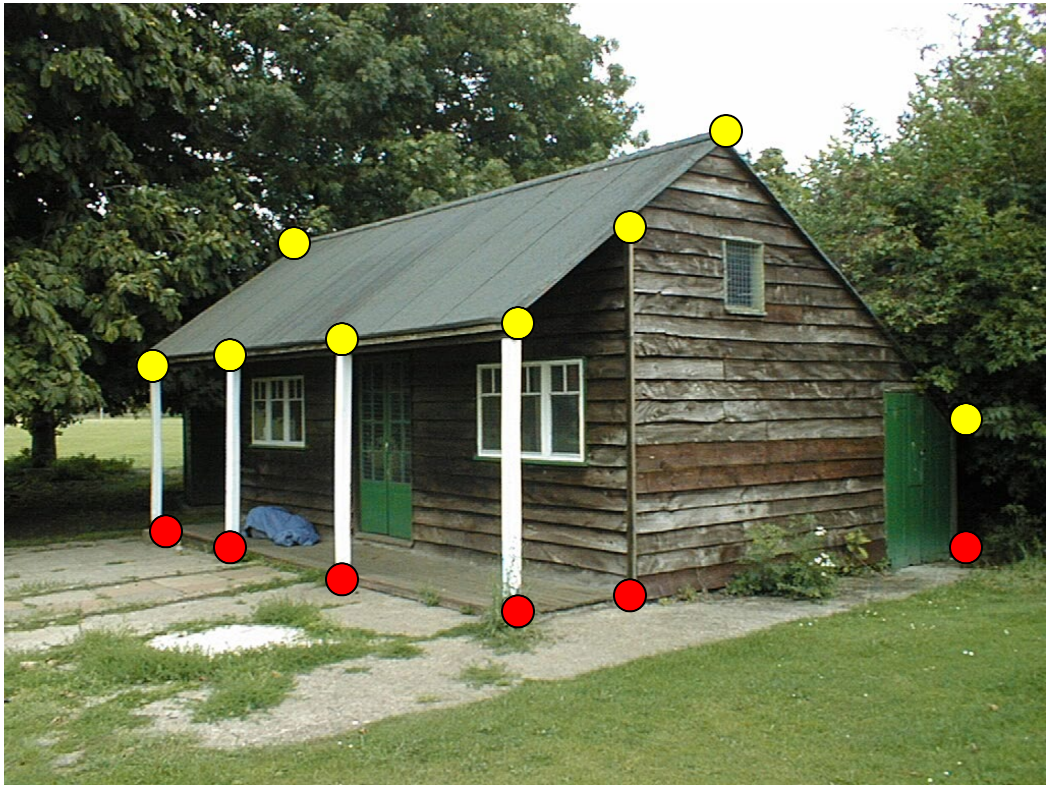
Our old friend – the homography

Need 4 reference points with world coordinates

$p = (x,y)$

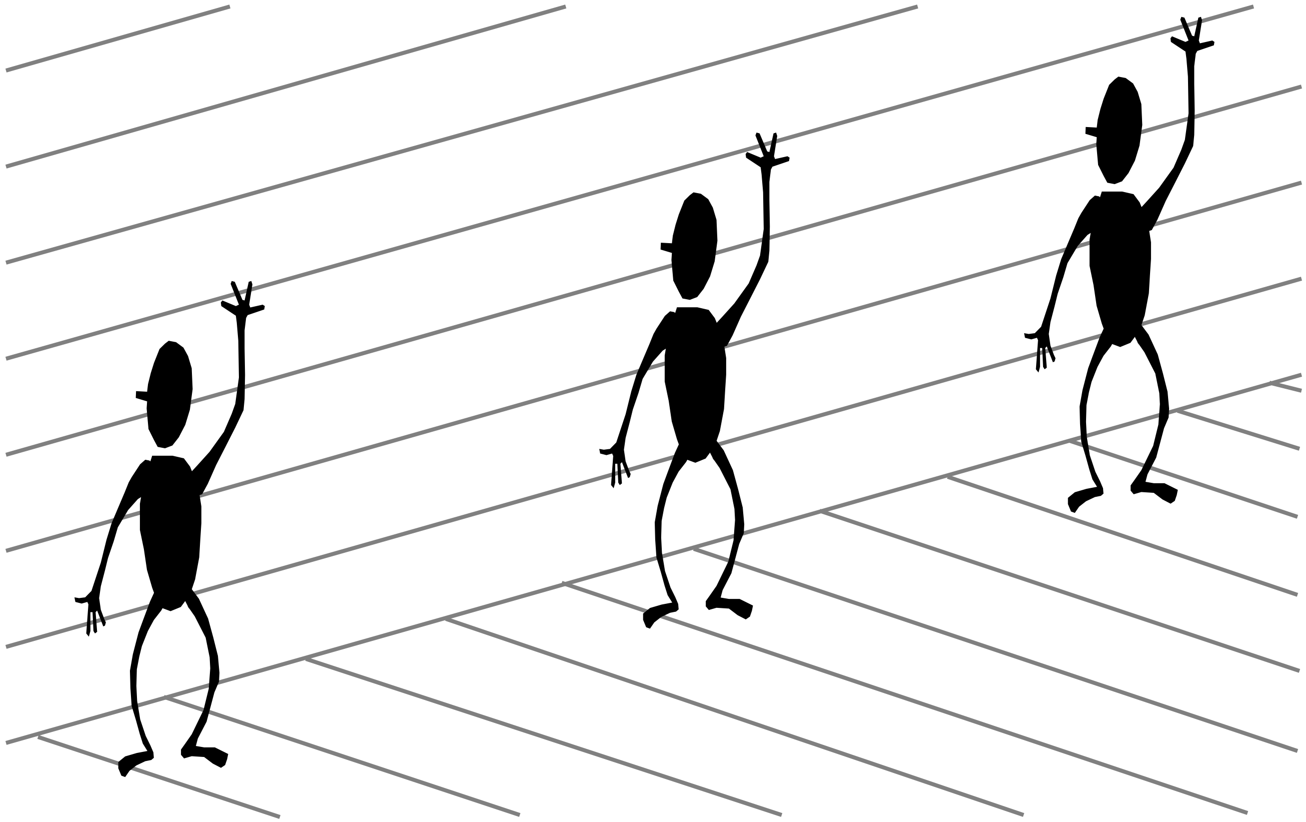
$p' = (X,Y,0)$

Finding world coordinates (X,Y,Z)

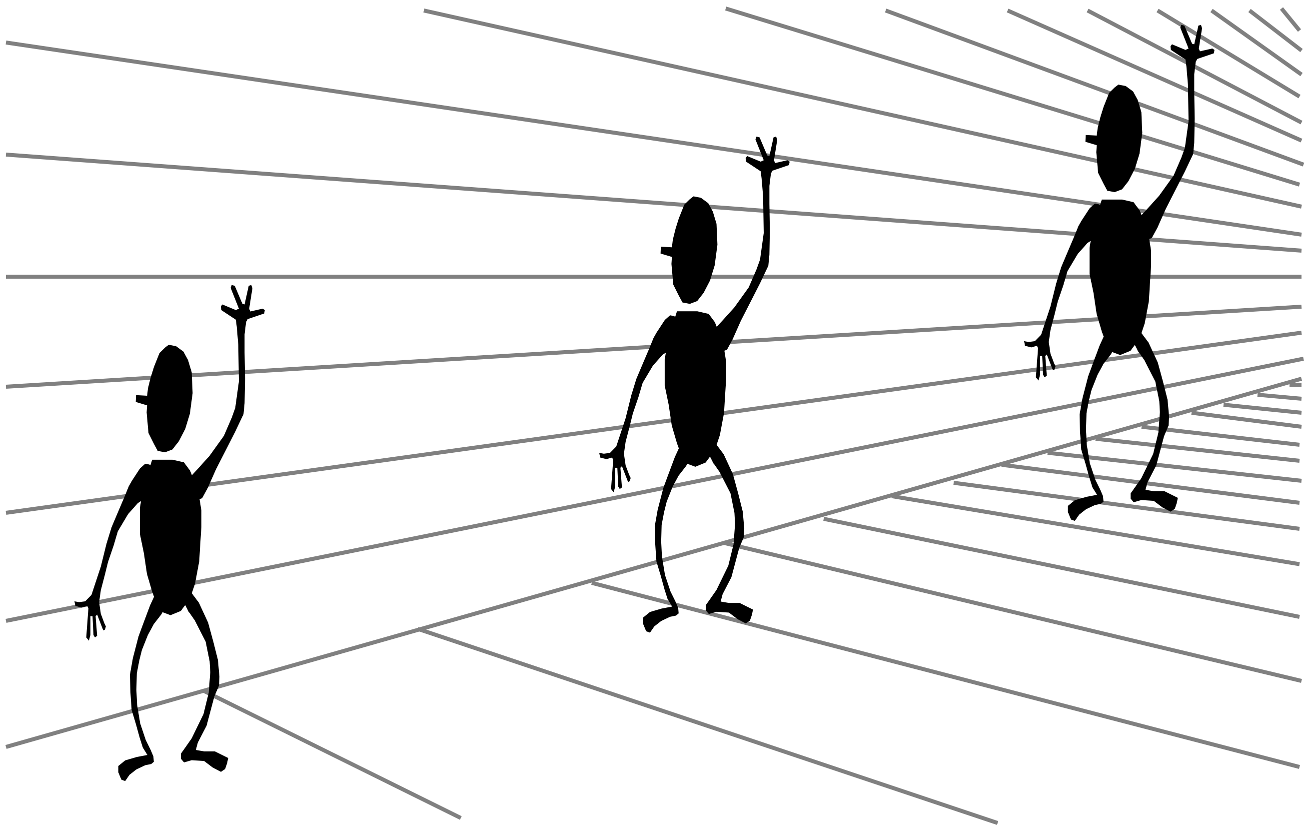


1. Define the ground plane (Z=0)
2. Compute points (X,Y,0) on that plane
3. Compute the *heights* Z of all other points

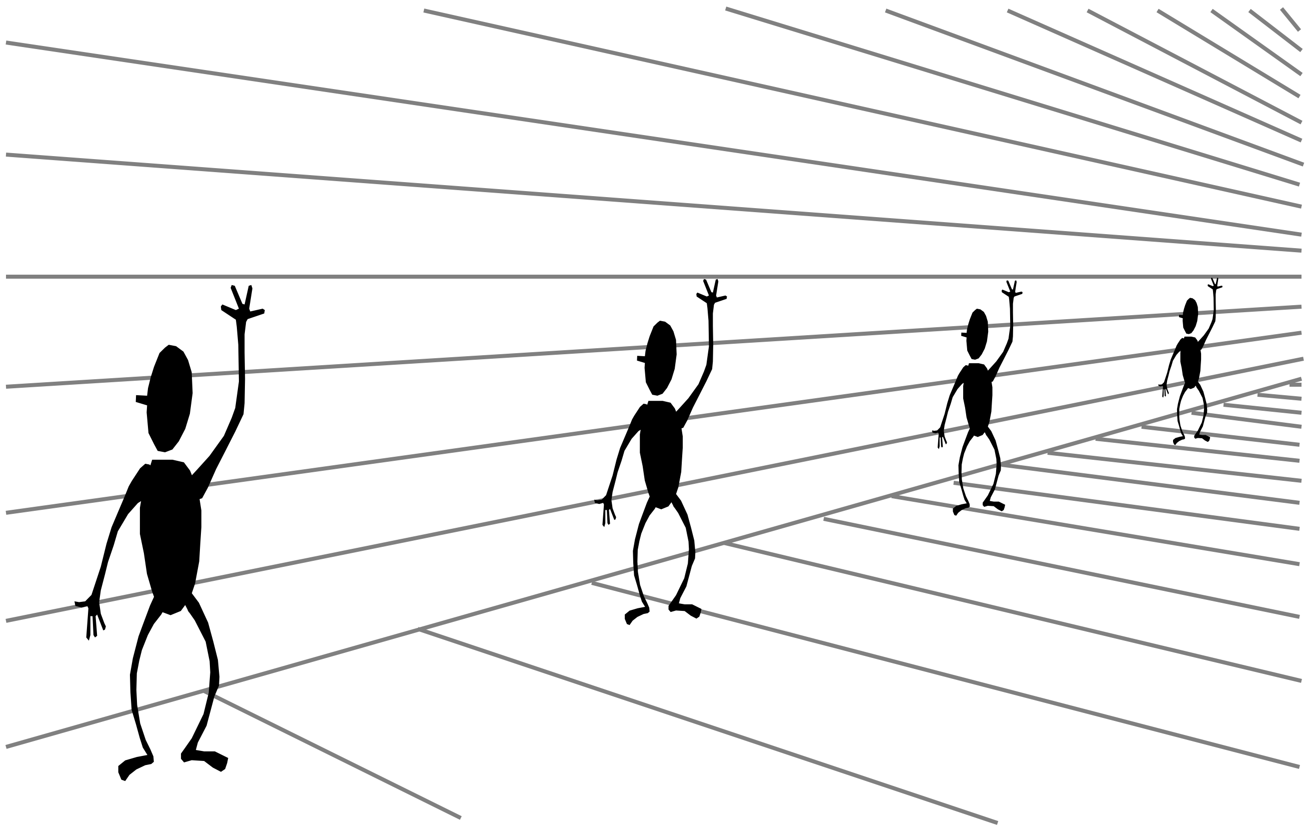
Comparing heights



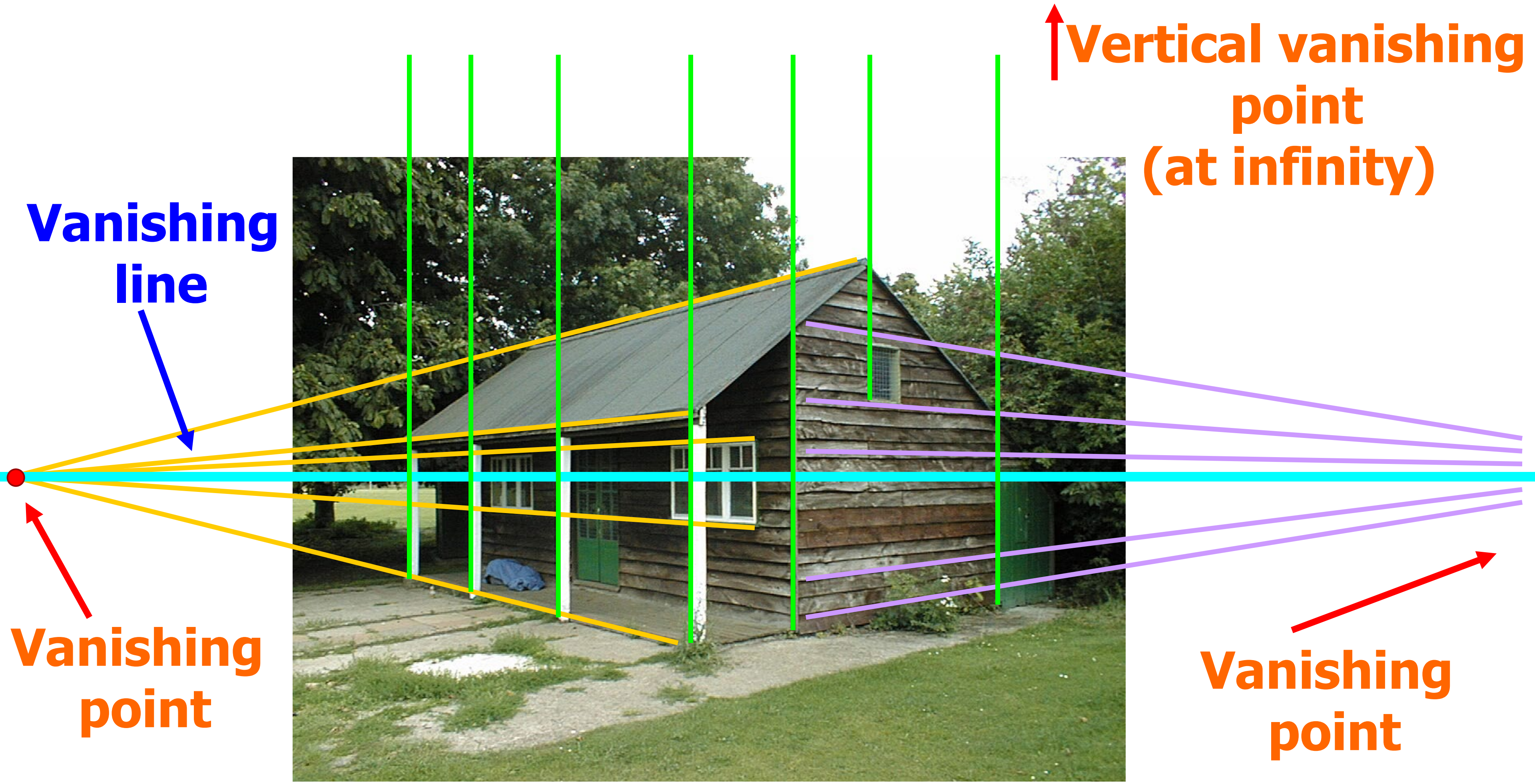
Perspective cues



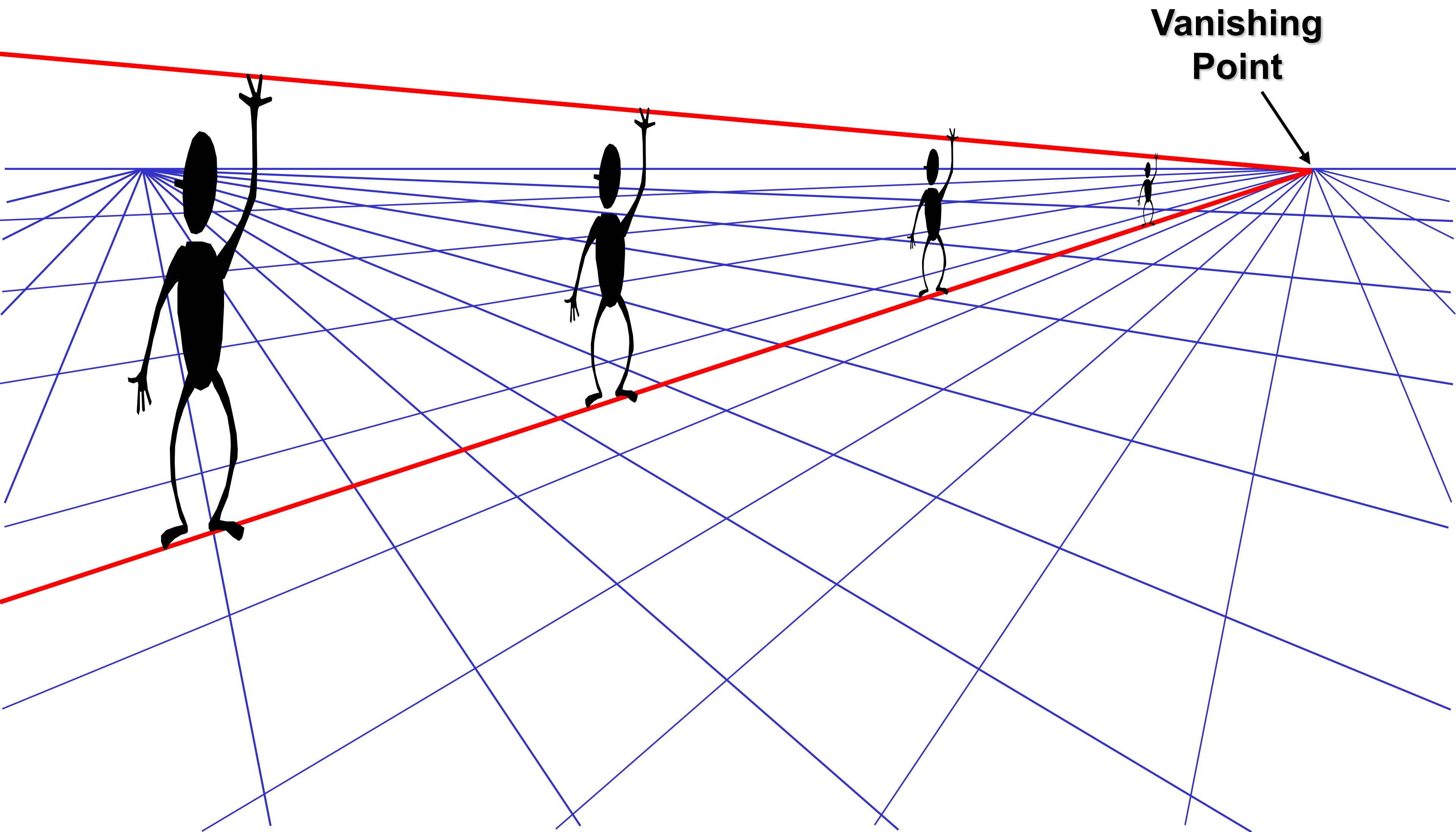
Perspective cues



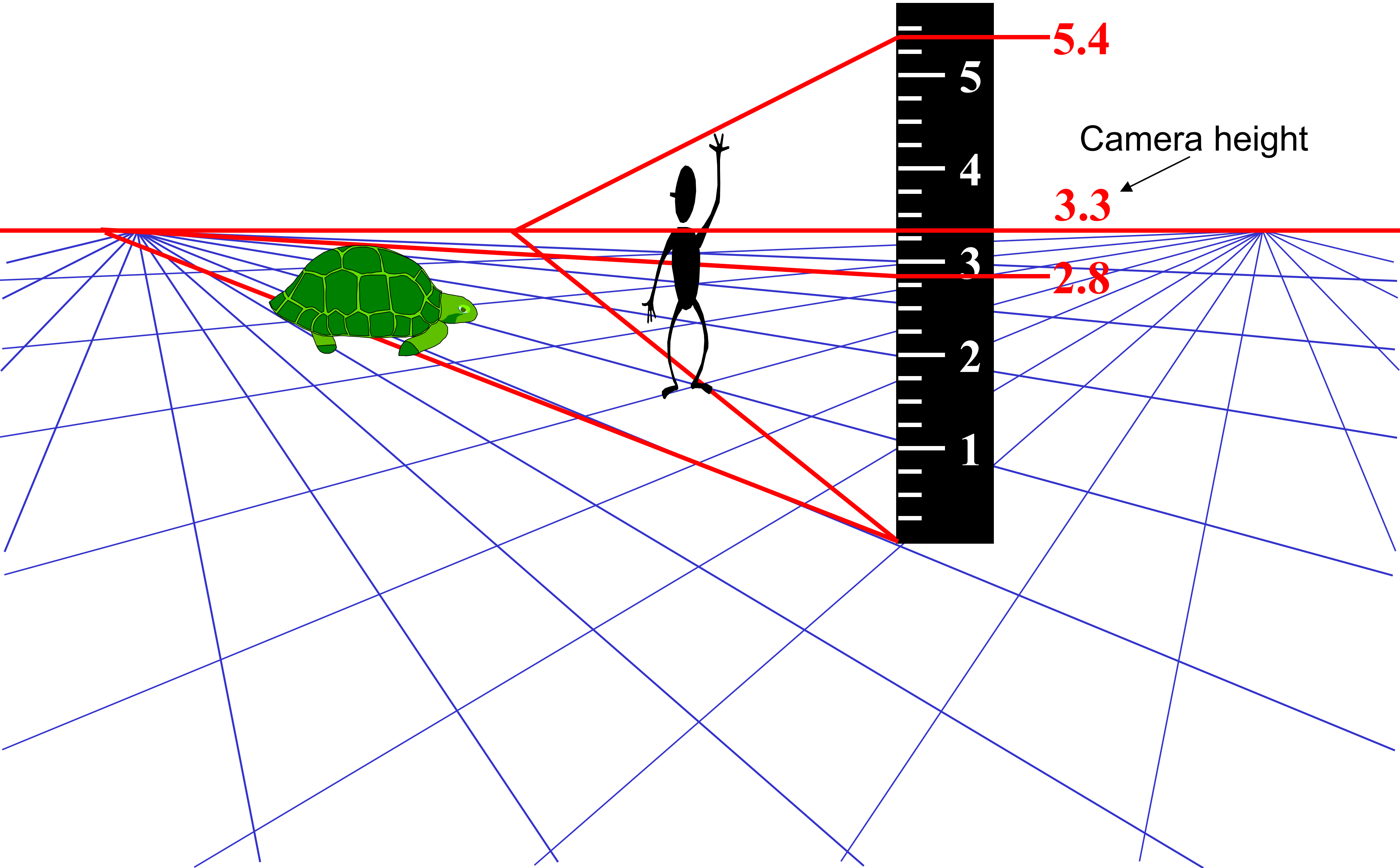
Criminisi ’99



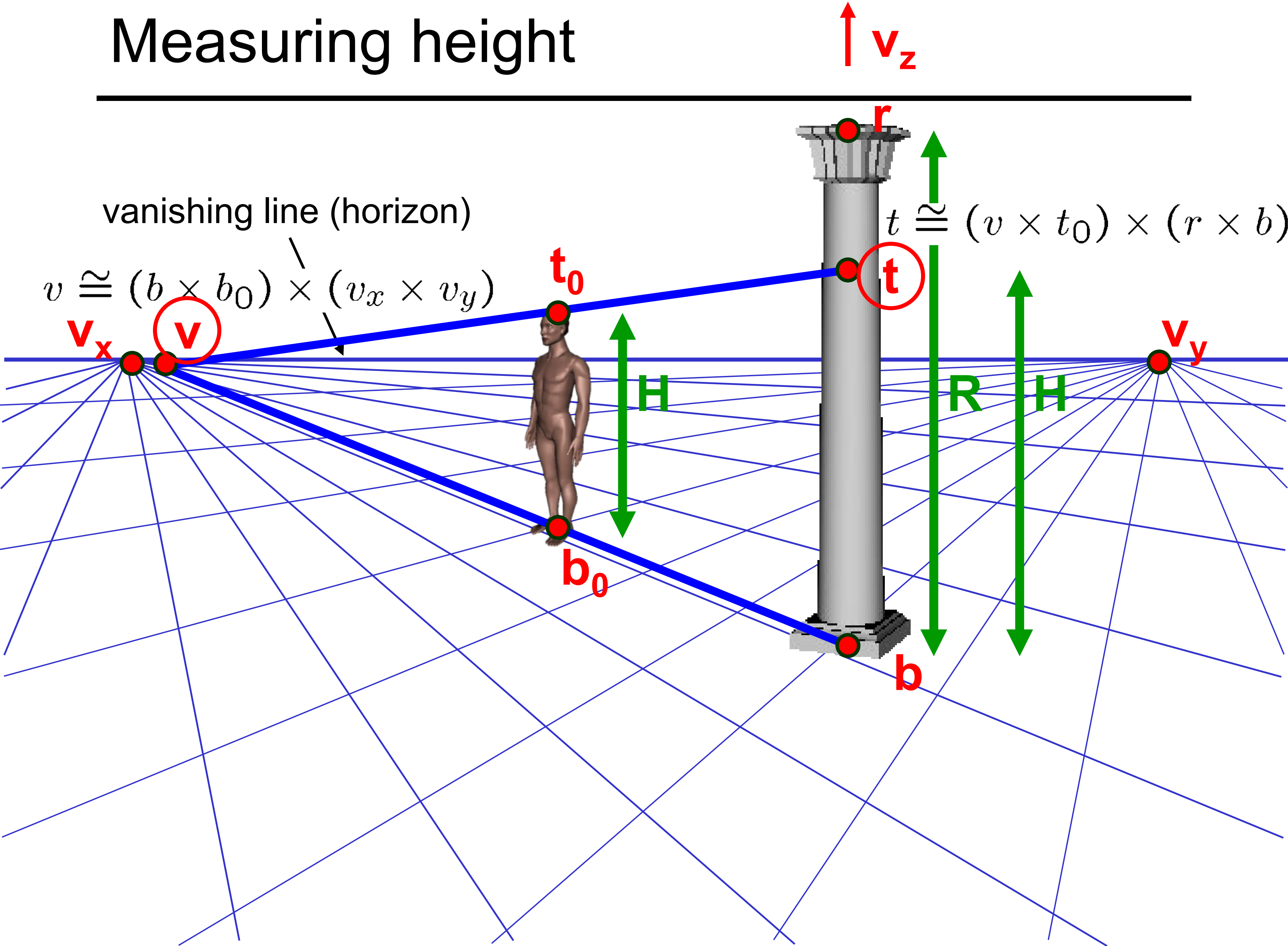
Comparing heights



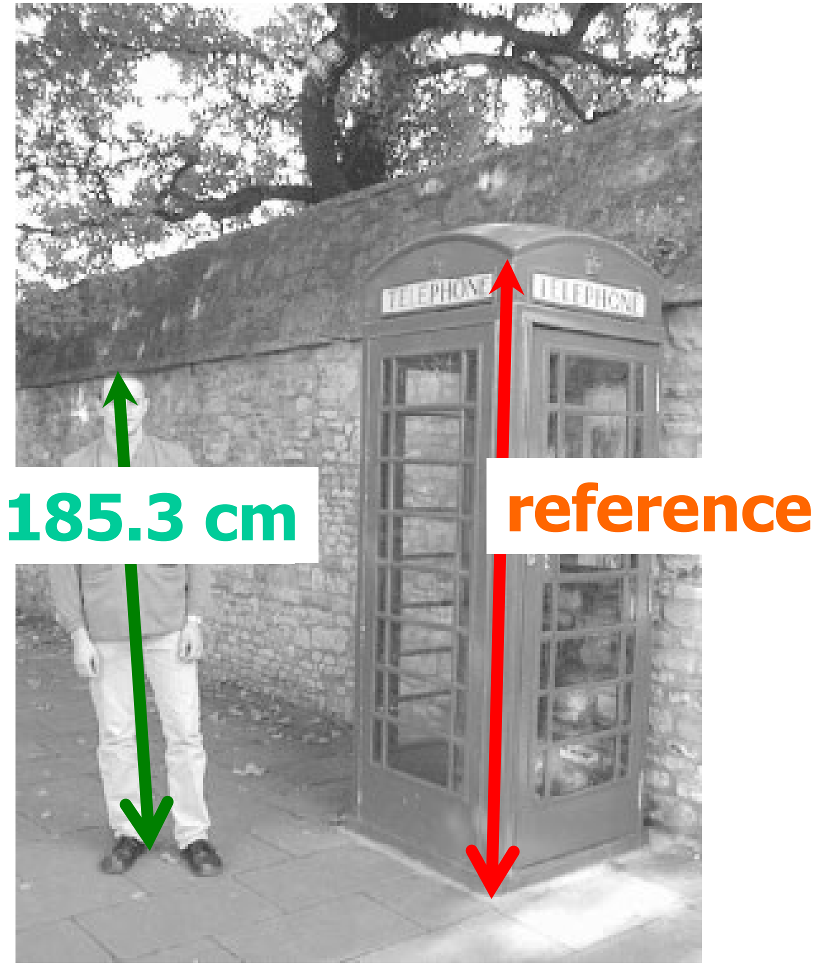
Measuring height



Measuring height



Measuring heights of people

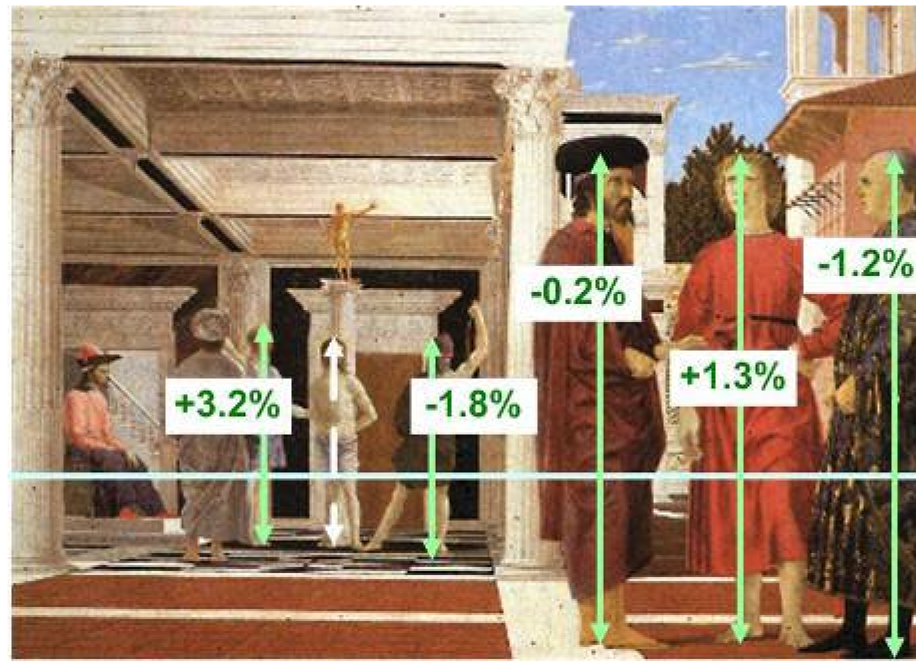


Here we go !

Assessing geometric accuracy

Are the heights of the 2 groups of people consistent with each other?

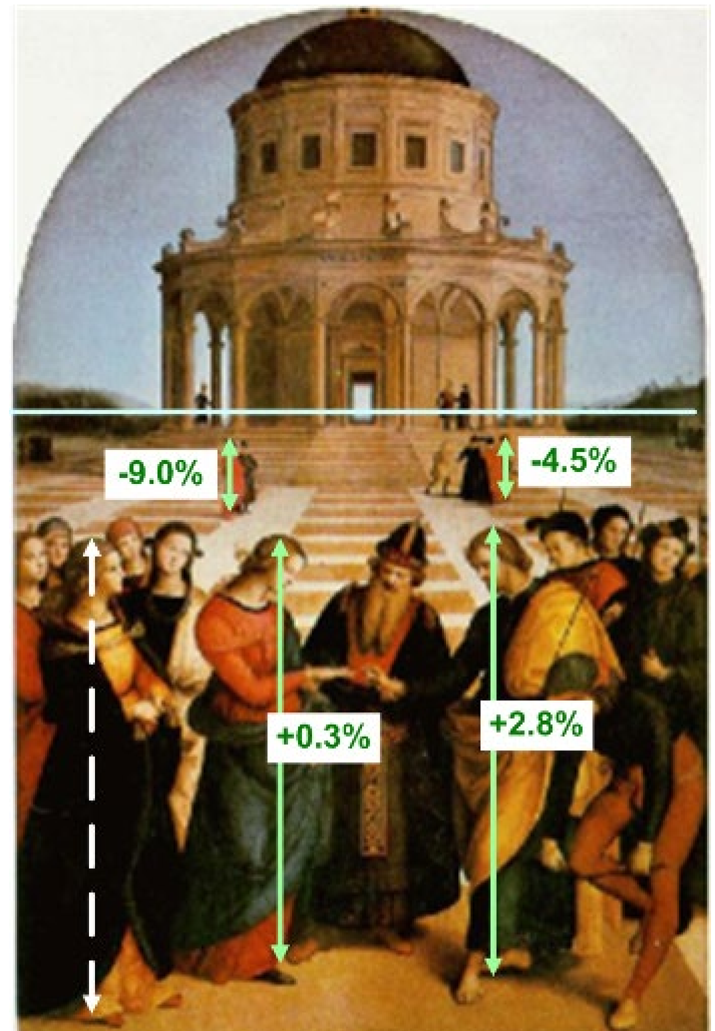
***Flagellation*,**
Piero della Francesca



Estimated relative heights

Assessing geometric accuracy

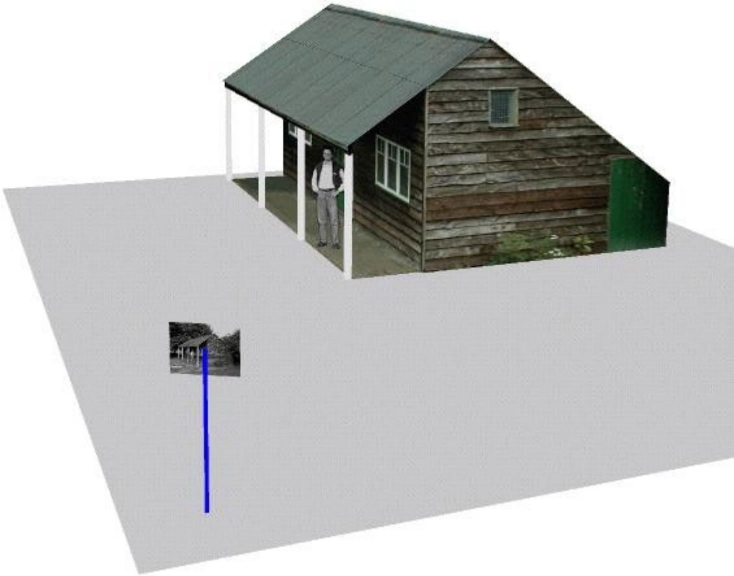
***The Marriage of the Virgin*, Raphael**



Estimated relative heights

Complete 3D reconstruction

Single View algorithms



Single image

- Planar measurements
- Height measurements
- Automatic vanishing point/line computation
- Interactive segmentation
- Occlusion filling
- Object placement in 3D model

3D model

A virtual museum @ Microsoft